EE219 Large-scale Data Mining

Course Project #3

Collaborative Filtering

Shijun Lu（905035448）

Wenfei Lu（505035450）

Xingyi Chen（205032924）

Hanren Lin（304944990）

Introduction

Recommender system, which is fundamentally based on the web technology, is becoming a trending concentration in various areas, especially in the entertainment-related websites. In this project, we applied collaborative filtering methods to the MovieLens dataset, in order to build a recommendation system for the prediction of the ratings of the movies from this dataset.

The collaborative filtering methods in this project mainly focused on two types of filtering: Neighborhood-based collaborative filtering, and Model-based collaborative filtering.

Part 1 Simple Analysis of MovieLens Dataset

(This part is for Question 1-6.)

The content of MovieLens Dataset is a "rating matrix" R with a size of m×n, with m representing users (rows) and n representing movies (columns). The (i, j) entry of the matrix represents the rating of user i for movie j.

In this part, we carried out simple analysis of MovieLens Dataset to visualize several properties of the dataset.

[Q1] First, we calculated the sparsity of the rating matrix R. The value of sparsity is the division of total number of available ratings by total number of possible ratings. The result is 0.000899038257317.

[Q2 – Q4] Second, we plotted 3 figures for further visualized analysis. The results are shown in Figure 1-3. Figure 1 is a histogram of the frequency of the rating values, showing an overall condition of the different rating's ratio. Figure 2 is the distribution of ratings among movies, and Figure 3 is the distribution of ratings among users.

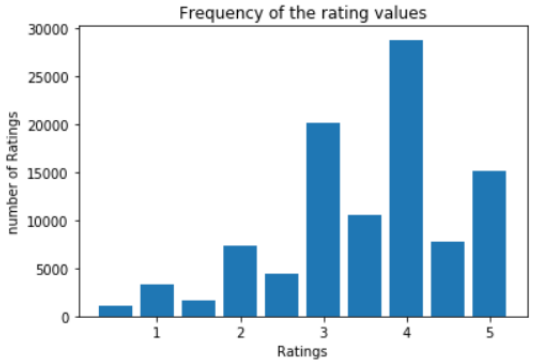


Figure 1 - Histogram of the frequency of the rating values

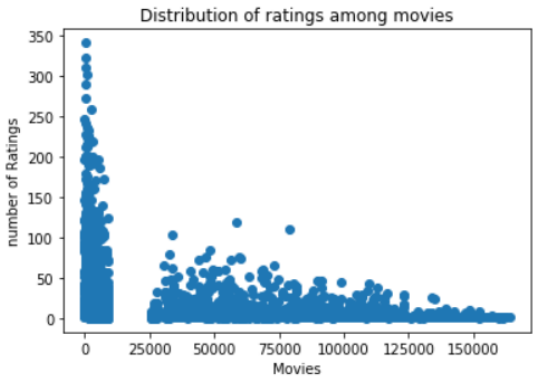


Figure 2 - Distribution of ratings among movies

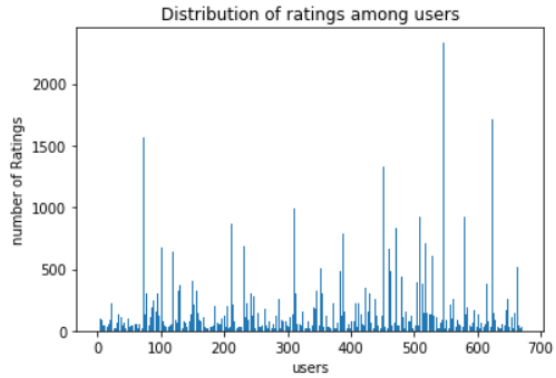


Figure 3 - Distribution of ratings among users

[Q5] From figure 2, a salient feature of the distribution in ratings among movies is that the number of the most rated movies occupies a small ratio with the index mostly between 0-10000, and the set of movies rated less than 100 users being the majority. When we build a recommendation system, we need to obtain rating information from other users or movies. The movies with most ratings can establish a solid reference when making prediction, while we still need to take movies with few ratings into consideration for better prediction result.

[Q6] We also calculated the variance of the rating values received by each movie. The horizontal axis represents the rating variance with intervals of width 0.5, and the vertical axis represents the number of movies with intervals of width 1000. The plotted result is shown in figure 4.

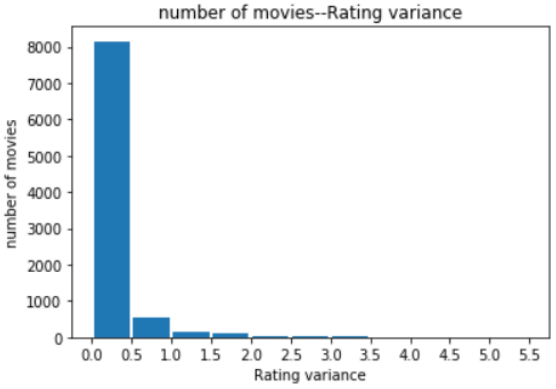


Figure 4 – The variance of the rating values received by each movie

From the histogram, the majority value of the variance is between 0~0.5, and when the variance gets higher, the overall percentage for corresponding variance would become smaller. In the later sections' analysis, we need to compensate the impact from the movies with high rating variance.

Part 2 Neighborhood-based Collaborative Filtering

(This part is for Question 7-15.)

Neighborhood-based collaborative filtering is a filtering method via matching various users according to their features' similarity, and making a prediction from the matching result of rating matrix R. We implemented user-based models, in which similar users have similar ratings on the same item.

In this filtering method, in order to determine the target user u's neighborhood, we need to calculate the u's similarity to all the other users. We used Pearson-correlation coefficient to build a similarity

function for the similarity calculation. Pearson-correlation coefficient (denoted by Pearson (u, v)) defines the similarity between the rating vectors of users u and v.

[Q7] Assume:

I_u: Set of item indices for which ratings have been specified by user u

I_v: Set of item indices for which ratings have been specified by user v

μ_u: Mean rating for user u computed using her specified ratings

r_{uk}: Rating of user u for item k

Then we can calculate μ_u with given I_u and r_{uk}:

$$\mu_u = \frac{\sum_{(u,k)\in I_u} r_{uk}}{|I_u|}$$

[Q8] The meaning of $I_u \cap I_v$ can be defined as the intersection set of user u and v's rated items. When $I_u \cap I_v = \emptyset$, it means there are no common rated items among user u and v. For a sparse matrix with a huge size like the rating matrix R, r_{uk} has a high possibility to have a value of zero, because one user would usually rate several movies that are related to personal interest. Using an intersection set could be an efficient way to target and match users with high similarity.

The prediction function, which evaluates the predicted rating of user u for item j, denoted by $\hat{r}_{uj}$, is given by the equation

$$\hat{r}_{uj} = \mu_u + \frac{\sum_{v\in P_u} Pearson\ (u,v)(r_{vj} - \mu_v)}{\sum_{v\in P_u} |Pearson\ (u,v)|}$$

[Q9] In this prediction function, $(r_{vj} - \mu_v)$ is a procedure of mean-centering the raw ratings. Mean-centering can be regarded as a measure for avoiding significant impact caused by several extreme data samples. If one user (v) gives severely biased ratings on all the movies, such as all positive rating or negative rating, the values of all r_{vj} would be high or low, with μ_v having a similarly value. Using a factor $(r_{vj} - \mu_v)$ can leave out these extreme samples, since for these samples the mean-centering factor's value would be relatively small.

[Q10] In this part, first, we used a self-designed k-NN collaborative filter for the prediction of the movie's ratings, and evaluated the performance via 10-fold cross validation. The value of k (number of neighbors) was swept from 2 to 100 in step sizes of 2, with respectively calculated values average RMSE and MAE across all 10 folds. Figure 5 shows the result of average RMSE versus k, and Figure 6 shows the result of average MAE versus k.

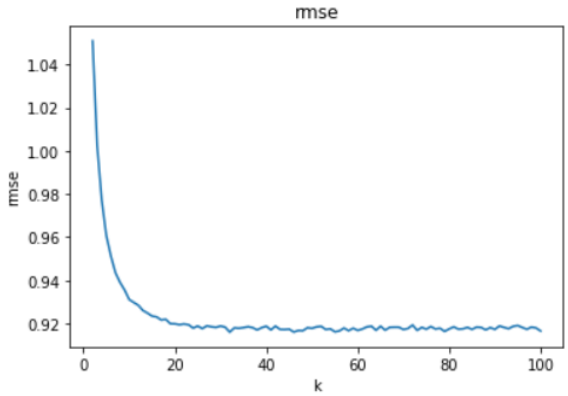


Figure 5 – k-NN, The result of average RMSE versus k

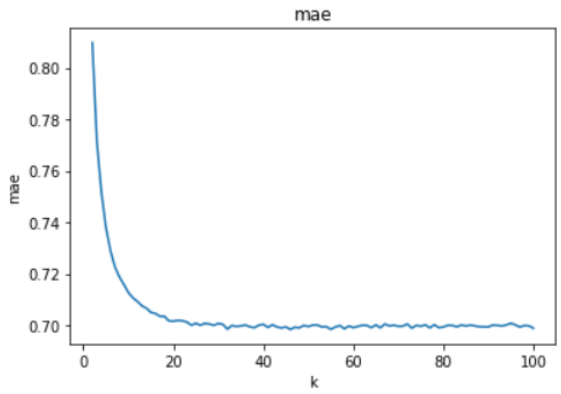


Figure 6 – k-NN, The result of average MAE versus k

[Q11] In addition, a 'minimum k' was found from the Figures above. The 'minimum k' refers to the value of k when the value of RMSE and MAE turning into a steady state. From the plot, the value of 'minimum k' is 35.

[Q12 – Q14] Second, the performance of the k-NN collaborative filter in predicting the ratings of the movies in the trimmed test set was also evaluated. The test set would be trimmed via three methods in this project: Popular movie trimming, unpopular movie trimming, and high variance movie trimming. With the self-designed k-NN collaborative filter, we predicted the ratings of the movies from three movies test set with different trimming method.

The value of k was swept in the same way as that in 10-fold cross validation. The results of average RMSE versus k are plotted in Figure 7, 8, 9, with the minimum average RMSE reported in Form 1.

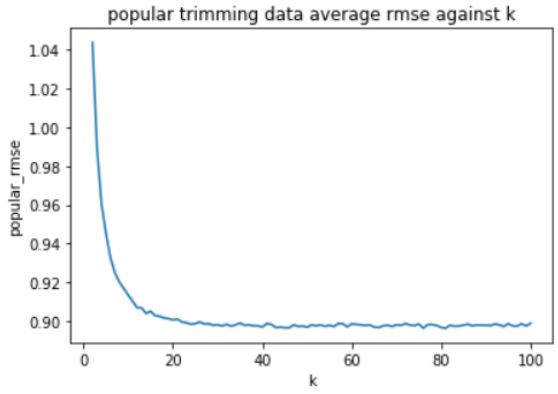


Figure 7 – k-NN, Average RMSE versus k (Popular Trimming Data)

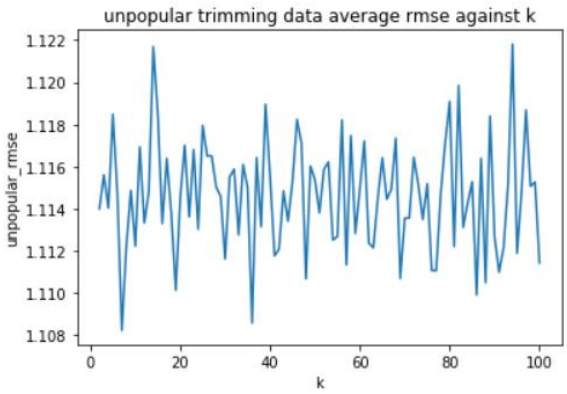


Figure 8 – k-NN, Average RMSE versus k (Unpopular Trimming Data)

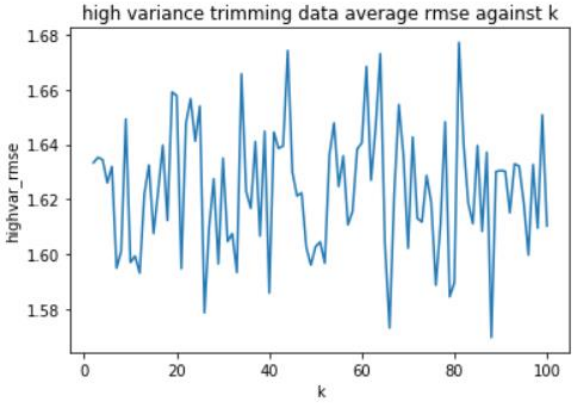


Figure 9 – k-NN, Average RMSE versus k (High Variance Trimming Data)

Type of Trimmed Test Set	Minimum Average RMSE	Optimal k
Popular	0.896537	61
Unpopular	1.107771	28
High Variance	1.564669	79

Form 1 – k-NN, Minimum average RMSE and Optimal k for Each Test Set

[Q15] In addition, we used ROC curve as an approach of filter performance evaluation. From project 1, we've acknowledged that ROC curve provides an efficient way for the visualization of binary classifier's performance. In this project, to adjust the rating recommendation system with a continuous scale into a binary system, we can set a threshold to divide the rating scores into two categories. The rating scores below the threshold would be set to 0, and vice versa.

We applied 4 different threshold values for this question: 2.5, 3, 3.5, 4. In this question, the value of k is the previously mentioned 'minimum k'. The ROC curves with area under the curve (AUC) value are shown in Figure 10 - 13.

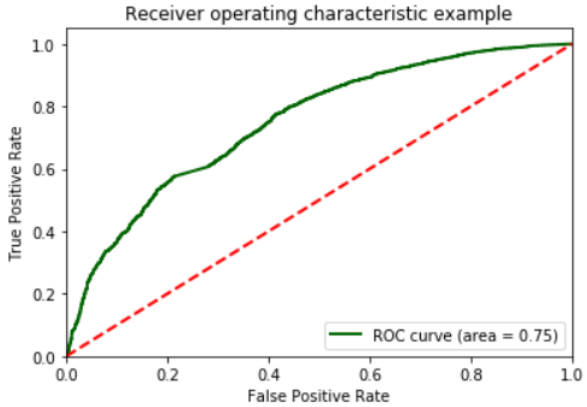


Figure 10 – k-NN, ROC Curve with threshold = 2.5 (AUC Value = 0.75)

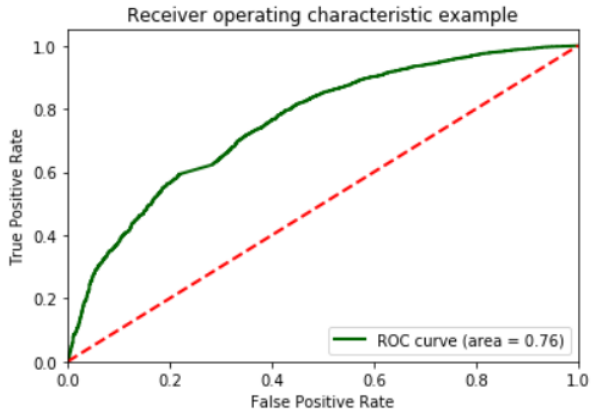


Figure 11 – k-NN, ROC Curve with threshold = 3 (AUC Value = 0.76)

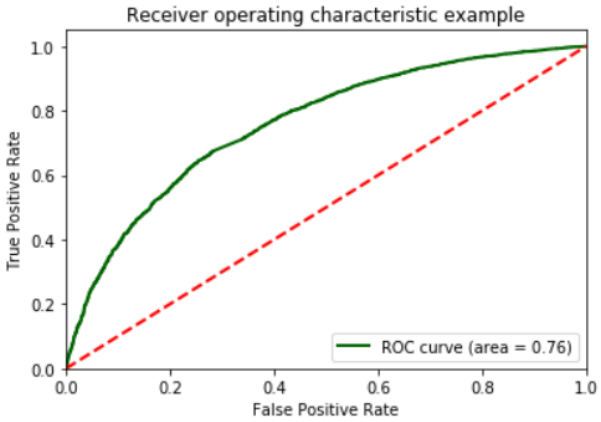


Figure 12 – k-NN, ROC Curve with threshold = 3.5 (AUC Value = 0.76)

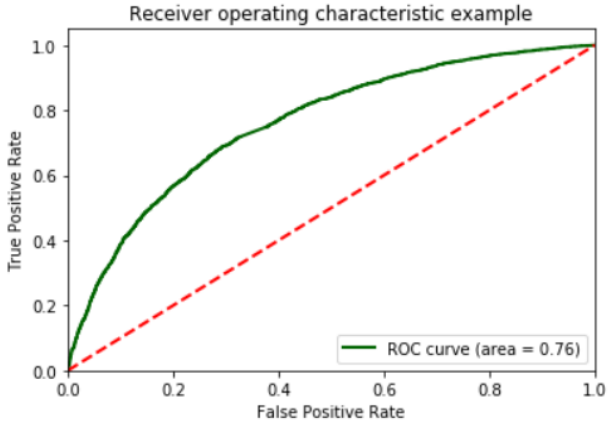


Figure 13 – k-NN, ROC Curve with threshold = 4 (AUC Value = 0.76)

Part 3 Model-based Collaborative Filtering

(This part is for Question 16-29.)

The models in model-based collaborative filtering are developed based on machine learning algorithms to predict users' rating of unrated items. For the project purpose, we adapted latent factor

based models for collaborative filtering.

Latent factor based models focus on estimation of missing entries from existing information in an incomplete matrix. The result of rating matrix R's approximation is a low-rank matrix with a robust estimation. To complete the matrix's approximation, we need to finish unconstrained matrix factorization formulation. In this project, two types of matrix factorization were explored: Non-negative matrix factorization (NNMF), and matrix factorization with bias (MF with bias).

Part 3.1 Non-negative Matrix Factorization (NNMF)

[Q16] For the optimization problem of latent factor based collaborative filtering, it could be reformulated as:

$$minimize_{U,V} \sum_{i=1}^{m} \sum_{j=1}^{n} W_{ij}(r_{ij} - (UV^T)_{ij})^2$$

The subject of this optimization problem is the function $f = \sum_{i=1}^{m} \sum_{j=1}^{n} W_{ij}(r_{ij} - (UV^T)_{ij})^2$, which includes square calculation. Therefore, this optimization problem can be regarded as convex optimization.

[Q17] In this part, first, we designed a NNMF-based collaborative filter and evaluated its performance via 10-fold cross validation. The value of k (number of latent factors) was swept from 2 to 50 in step sizes of 2, and each value of k corresponds to a calculation result of the average RMSE and average MAE across all 10 folds. The plotted results for the average RMSE versus k and the average MAE versus k are shown in Figure 14, 15.

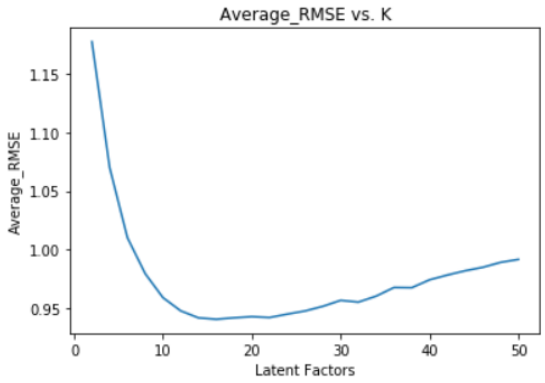


Figure 14 – NNMF-based, The result of average RMSE versus k

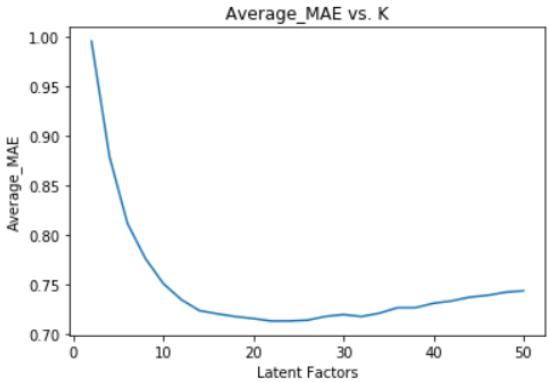


Figure 15 – NNMF-based, The result of average MAE versus k

[Q18] The optimal number of latent factors can be obtained when the value of average RMSE or average MAE is minimized. From the figures, when evaluating average RMSE, the optimal value of latent factors is 16. When evaluating average MAE, the optimal value of latent factors is 22. The value of the minimum average RMSE is 0.940345, and the value of minimum average MAE is 0.713612.
Latent factors evaluate the number of columns in latent factor matrix V, which means this optimal latent factor value can be regarded as the number of genres. Further details would be demonstrated in Q23.

[Q19 – Q21] Next, we applied the self-designed NNMF-based collaborative filter to the trimmed test sets and evaluated the performance of filtering. The operation procedure is similar with that in part 2, except the subject of sweeping being number of latent factors, and sweeping range being from 2 to 50 in this part. The plotted results of average RMSE versus k in three trimmed test sets are shown in Figure 16-18, and the values of minimum average RMSE in three test sets with corresponding optimal k are listed in Form 2.

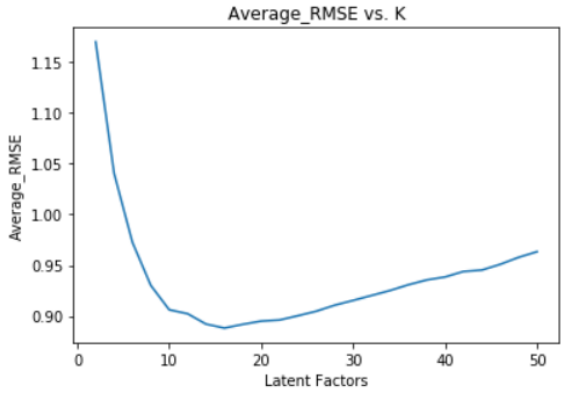


Figure 16 – NNMF-based, Average RMSE versus k (Popular Trimming Data)

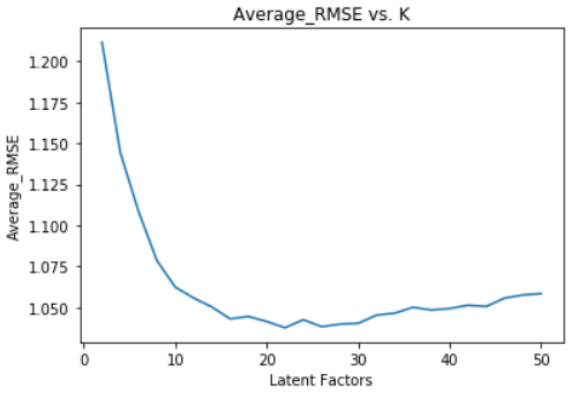


Figure 17 – NNMF-based, Average RMSE versus k (Unpopular Trimming Data)

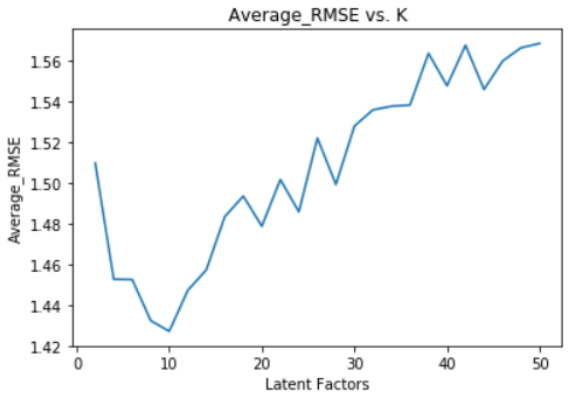


Figure 18 – NNMF-based, Average RMSE versus k (High Variance Trimming Data)

Type of Trimmed Test Set	Minimum Average RMSE	Optimal k
Popular	0.888225	16
Unpopular	1.037476	22
High Variance	1.427234	10

Form 2 – NNMF-based, Minimum average RMSE and Optimal k for Each Test Set

[Q22] We also used ROC curve for the evaluation with the same operation procedure and threshold values in part 2. The ROC curve results are plotted in Figure 19-22, with corresponding AUC value reported.

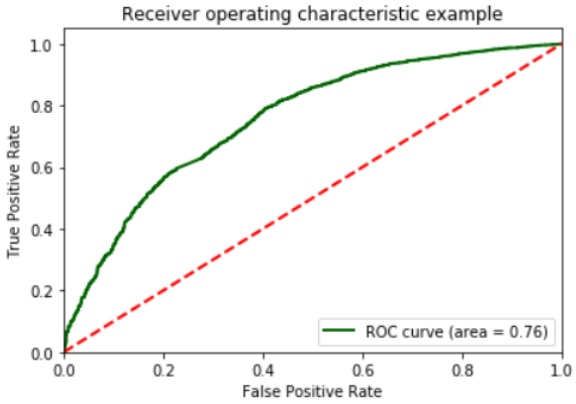


Figure 19 – NNMF-based, ROC Curve with threshold = 2.5 (AUC Value = 0.76)

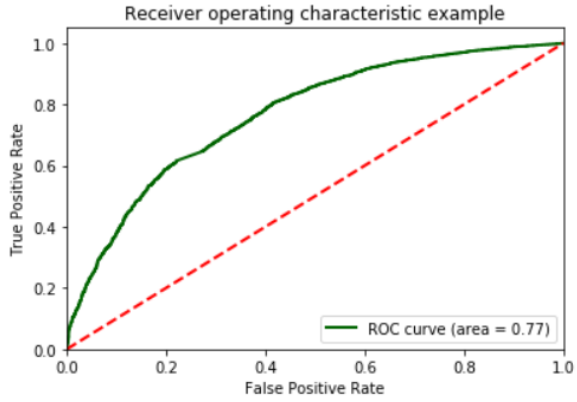


Figure 20 – NNMF-based, ROC Curve with threshold = 3 (AUC Value = 0.77)

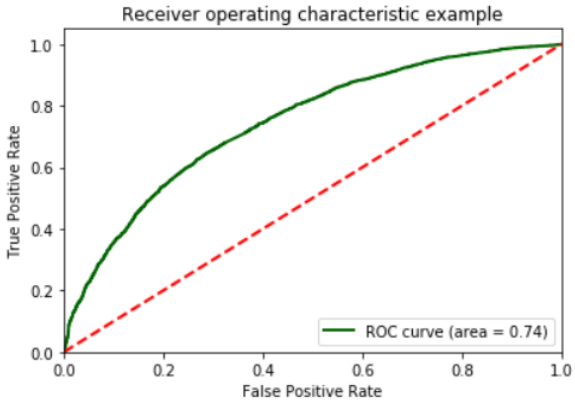


Figure 21 – NNMF-based, ROC Curve with threshold = 3.5 (AUC Value = 0.74)

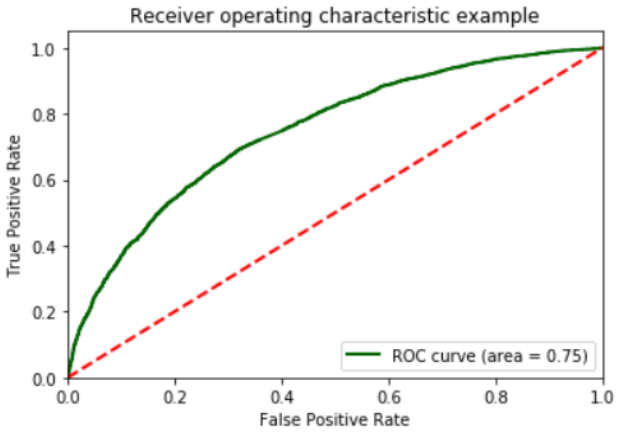


Figure 22 – NNMF-based, ROC Curve with threshold = 4 (AUC Value = 0.75)

[Q23] In addition, we performed non-negative matrix factorization on the rating matrix R to obtain the latent factor matrices U and V. For the matrix V, the genres of the top 10 movies were reported in Form 3, 4, 5.

Column 0	Column 1
Drama | Mystery | Romance	Comedy | Documentary
Action | Comedy | Crime | Fantasy	Adventure | Animation
Comedy | Documentary	Musical
Drama | War	Comedy | Romance
Children | Comedy	Drama | Fantasy | Horror
Action | Adventure | Sci-Fi | War | IMAX	Comedy
Thriller	Drama | Sci-Fi
Comedy | Western	Drama
Action | Adventure | Sci-Fi | IMAX	Adventure | Drama | Sci-Fi
Drama	Drama | Romance

Form 3 - Genres of the top 10 movies for column 0 & 1

Column 2	Column 3
Adventure | Drama | Fantasy | Romance	Drama
Action | Adventure | Drama | Thriller	Comedy | Crime

Action | War	Comedy | Drama
Action | Crime | Thriller	Documentary | Drama
Comedy	Action
Comedy | Drama | Romance	Drama | Thriller
Comedy	Adventure | Children
Comedy | Mystery | Thriller	Drama | Romance
Drama	Comedy
Comedy	Drama

Form 4 - Genres of the top 10 movies for column 2 & 3

Column 4
Comedy | Crime | Mystery | Thriller
Action | Comedy
Documentary
Action | Adventure | Sci-Fi | Thriller
Comedy | Drama
Drama | Romance
Children | Comedy | Musical | Romance
Comedy
Documentary
Comedy | Romance

Form 5 - Genres of the top 10 movies for column 4

From the results reported above, in each column, we can always distinguish several genres with significantly high frequency. In the column 0, “comedy”, “action” and “drama” are the most, while in the column 3 “thriller”, “comedy”, “drama” become high frequency genres. “Drama” and “comedy” both have high frequency in each column, which might be attributed to the fact that dramas and comedies are the genres with highest frequency in the complete dataset.
According to the attribute of the latent factor model, the responses on indicators or manifest variables are from the user’s preference of movies. Latent factor in rating matrix can reflect different user’s bias, which can be interpreted as a corresponding relation between the latent factors and movie genres. Therefore, we can draw a conclusion that the 20 latent factors are related to 20 collections of movie genres.

Part 3.2 Matrix Factorization with Bias (MF with bias)

[Q24] We designed a MF with bias collaborative filter and evaluated its performance via 10-fold cross validation. The procedure is similar with that in part 3.1. The plotted results for the average RMSE versus k and the average MAE versus k are shown in Figure 23, 24.

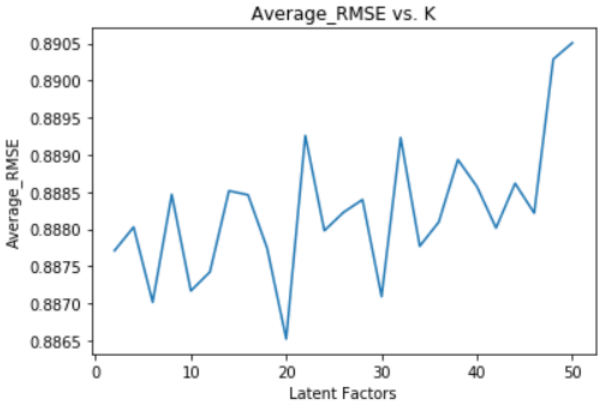


Figure 23 – MF with bias, The result of average RMSE versus k

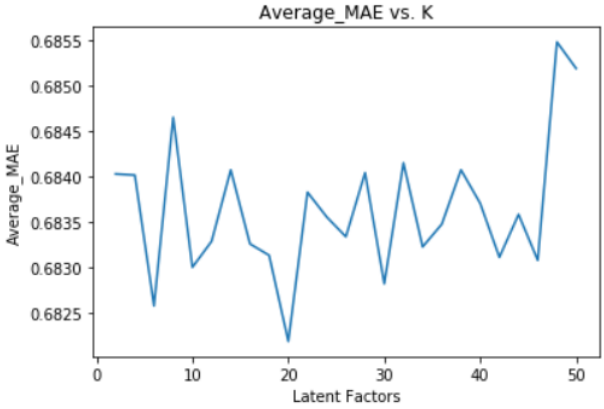


Figure 24 – MF with bias, The result of average MAE versus k

[Q25] The optimal number of latent factors can be obtained when the value of average RMSE or average MAE is minimized. From the figures shown above, the optimal value of latent factors is 20. The value of the minimum average RMSE is 0.886521, and the value of minimum average MAE is 0.682190.

[Q26 – Q28] Next, we applied the self-designed MF with bias collaborative filter to the trimmed test sets and evaluated the performance of filtering. After same operations as that in part 3.1, the plotted results of average RMSE versus k in three trimmed test sets are shown in Figure 25-27, and the values of minimum average RMSE in three test sets are listed in Form 6.

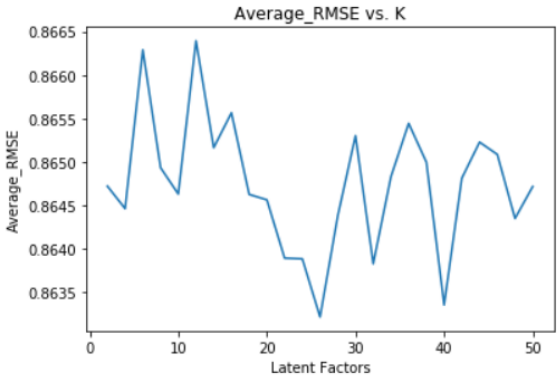


Figure 25 – MF with bias, Average RMSE versus k (Popular Trimming Data)

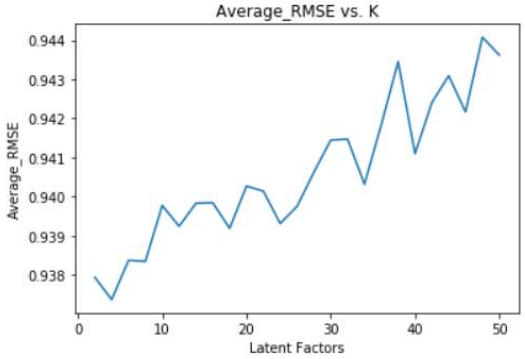


Figure 26 – MF with bias, Average RMSE versus k (Unpopular Trimming Data)

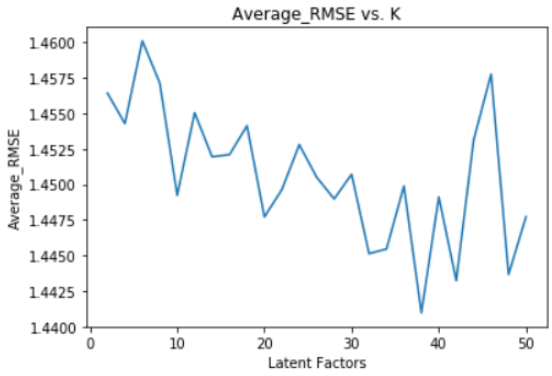


Figure 27 – MF with bias, Average RMSE versus k (High Variance Trimming Data)

Type of Trimmed Test Set	Minimum Average RMSE	Optimal k
Popular	0.863211	26
Unpopular	0.937360	4
High Variance	1.440978	38

Form 6 – MF with bias, Minimum average RMSE and Optimal k for Each Test Set

[Q29] We also used ROC curve for the evaluation with the same operation procedure and threshold values in part 2. The ROC curve results are plotted in Figure 28-31, with corresponding AUC value reported.

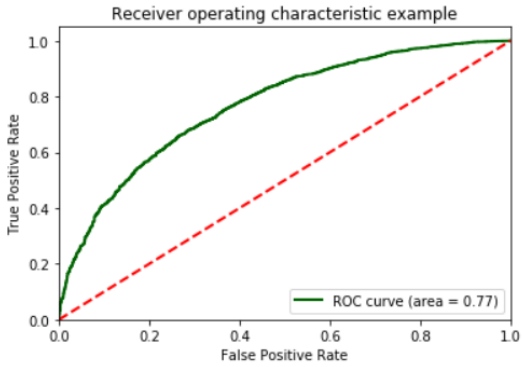


Figure 28 – MF with bias, ROC Curve with threshold = 2.5 (AUC Value = 0.77)

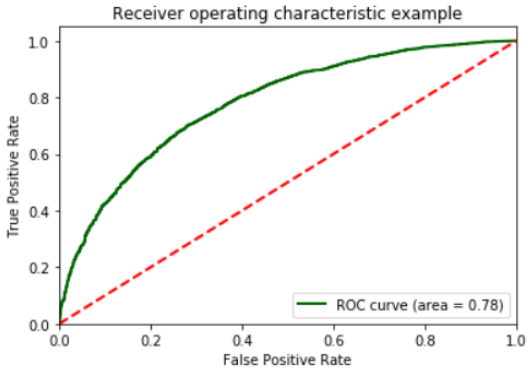


Figure 29 – MF with bias, ROC Curve with threshold = 3 (AUC Value = 0.78)

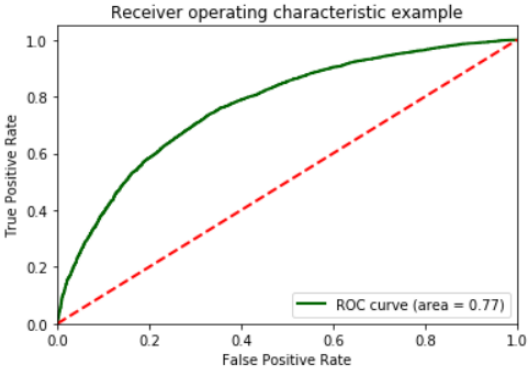


Figure 30 – MF with bias, ROC Curve with threshold = 3.5 (AUC Value = 0.77)

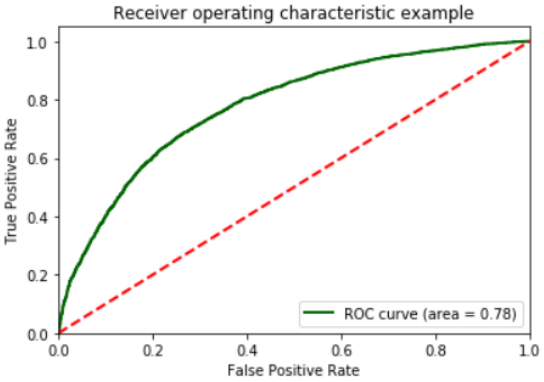


Figure 31 – MF with bias, ROC Curve with threshold = 4 (AUC Value = 0.78)

Part 4 Naïve Collaborative Filtering

(This part is for Question 30-33.)

Naïve collaborative filter returns the mean rating of the user as the predicted rating for an item, which means in this filter,

$$\hat{r}_{ij} = \mu_i$$

Where μ_i is the mean rating of user $\mathbf{i}$.

[Q30 – Q33] Cross-validation test and performance test on trimmed test set were conducted in this

part. The results with average RMSE reported are separately listed in Form 7, 8.

RMSE Results	0.9437	0.9732	0.9432	0.9521	0.9570	0.9683	0.9576	0.9623	0.9419	0.9545
Average RMSE	0.95538232560904712									

Form 7 - Result of Cross-validation Test

Trimmed Test Set	Average RMSE
Popular	0.954211864680
Unpopular	0.886063606432
High Variance	0.932509291971

Form 8 - Result of Performance on Trimmed Test Set

Part 5 Performance Comparison

(This part is for Question 34.)

[Q34] In this part, we compared the performance of k-NN, NNMF, and MF with bias collaborative filters by plotting the ROC curves in the same figure. The value of threshold is set to 3. The plotting results are shown in Figure 32.

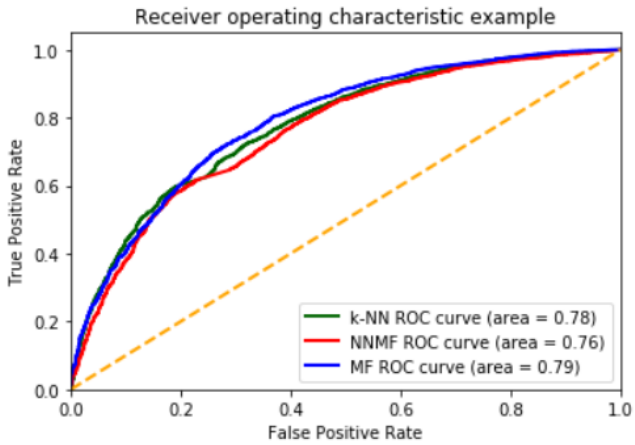


Figure 32 - ROC curve for 3 Collaborative Filters

From the figure shown above, for MF ROC curve, it has the largest AUC value, while the NNMF has the smallest AUC value. Therefore, MF with bias filter has the best performance, and k-NN filter has a slightly worse performance, with NNMF-based filter having a relatively worst performance.

Part 6 Ranking

(This Part is for Question 35-39.)

The previous parts focused on the prediction version of the problem. In this part, the concentration becomes the ranking version of the problem. We used precision-recall curve to evaluate the ranking.

[Q35] Precision evaluates an extent how "successful" the recommendation is. When a stationary set of items are recommended to a certain user, if there are more items that the user rates with "like" in this set, the precision would be higher. Recall evaluates an extent how "sensitive" the recommendation is. If the recommendation system can accurately provide more items that the user likes, the value of

recall would get higher. The mathematical expressions for precision and recall are given by:

$$Precision(t) = \frac{|S(t) \cap G|}{|S(t)|}$$

$$Recall(t) = \frac{|S(t) \cap G|}{|G|}$$

Where S(t) represents the set of items of size t recommended to the user, and G represents the set of items liked by the user.

[Q36 – Q38] First, we plotted the average precision versus t, average recall versus t, and average precision versus recall figures for the k-NN, NNMF-based, and MF with bias collaborative filter separately. The plotted results are shown in figure 33, 34, 35.

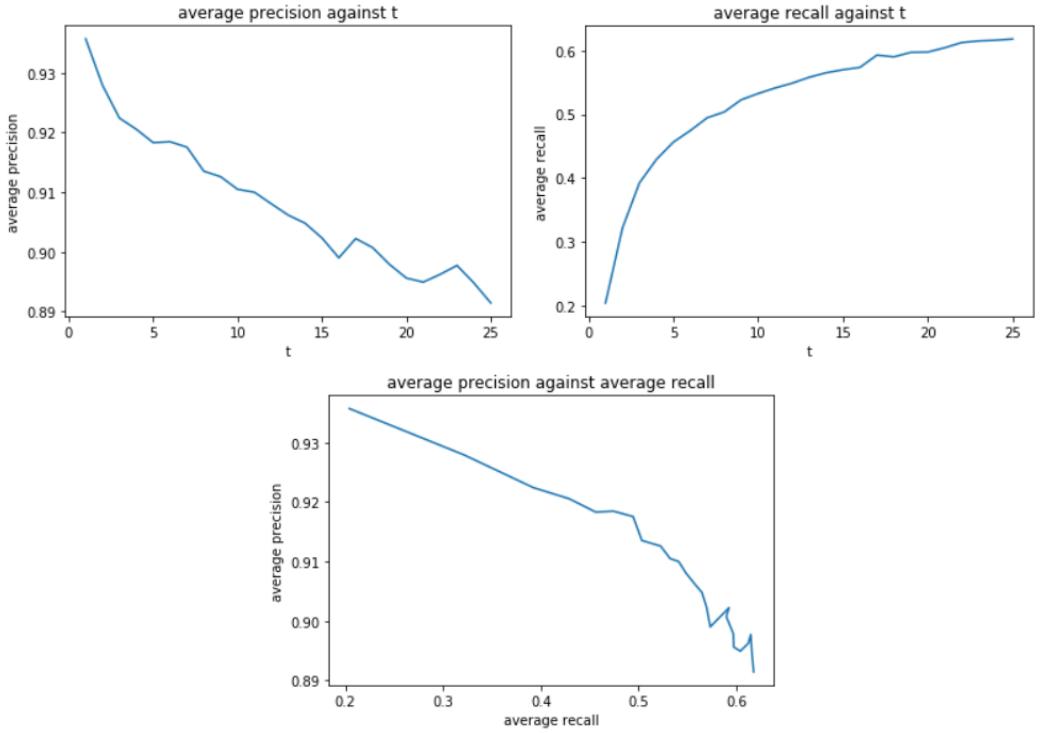
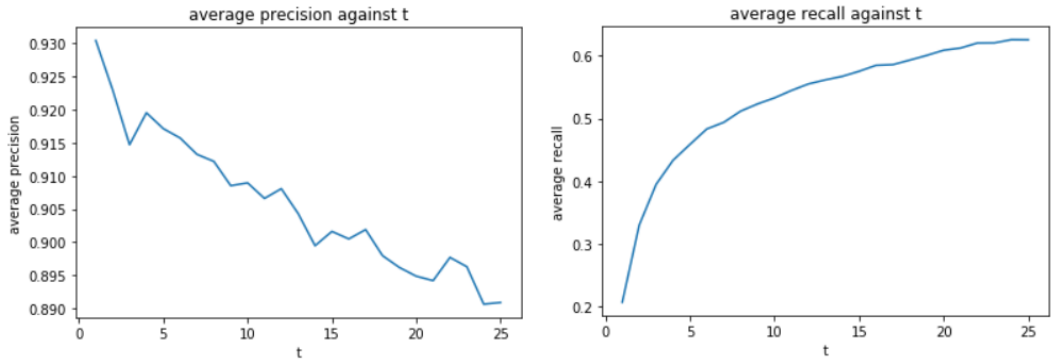


Figure 33 - 3 figures for k-NN collaborative filter



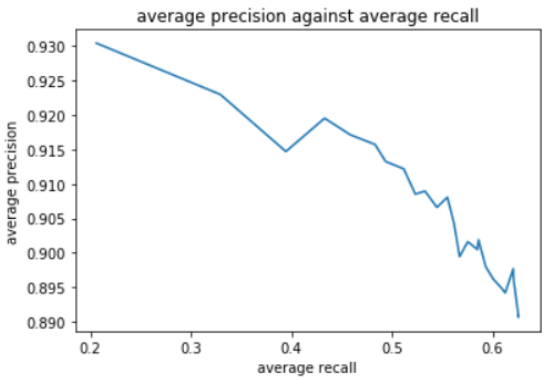


Figure 34 - 3 figures for NNMF-based collaborative filter

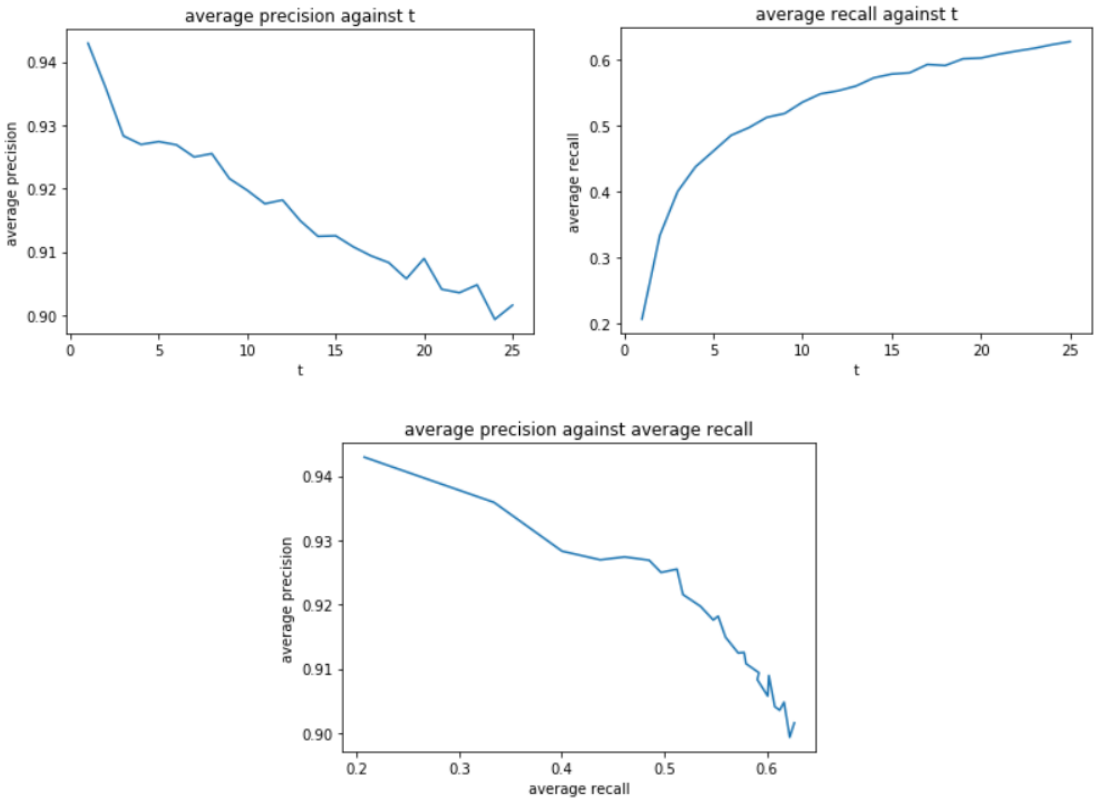


Figure 35 - 3 figures for MF with bias collaborative filter

From the figures shown above, the different figure has a unique tendency. For the average precision versus t curve, the overall tendency is rough decreasing, but approximately linear. For the average recall versus t curve, it's monotonically increasing, but the slope would decrease with t increasing, which is close to the feature of logarithm curve. For the average precision versus average recall curve, though the curve shows a tendency of decreasing, the extent of smoothness changes in different part of the curve. The first half has a small average slope with relatively smooth curve, nonetheless, the second half has a bigger average slope, with relatively rough curve.

[Q39] Second, we merged the 9 separated figures into 3 figures, with 3 filters' curves included in one corresponding figure. The plotted results are shown in figure 36.

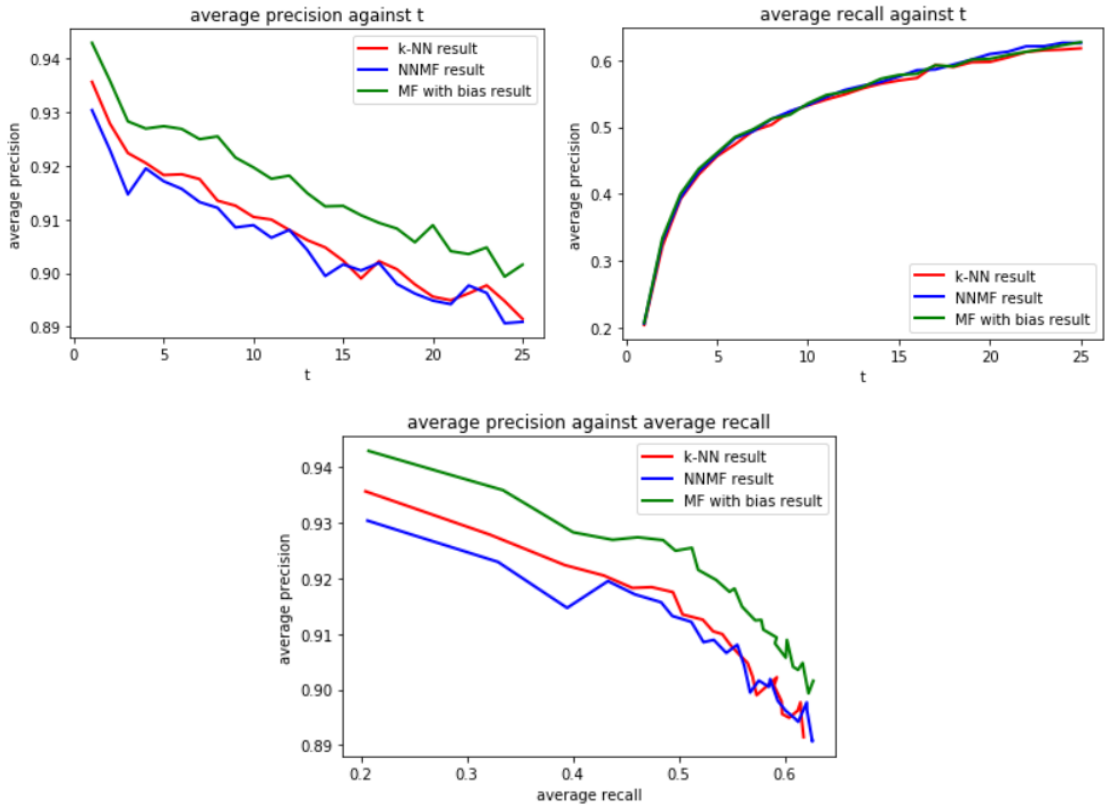


Figure 36 - 3 figures for all 3 kinds of collaborative filters